2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 14-18, 2024. Abu Dhabi, UAE

FI-SLAM: Feature Fusion and Instance Reconstruction for Neural Implicit SLAM

Xingshuo Wang[1], Yunzhou Zhang[1*], Zhiyao Zhang[1], Mengting Wang[1], Zhiteng Li[1], Xuanhua Chen[1]

***Abstract*— Recent advancements in neural implicit fields for Simultaneous Localization and Mapping (SLAM) have provided breakthroughs. However, the benefits of reconstruction results to the perception ability of robot are minimal. Therefore, we propose FI-SLAM, a dense semantic instance SLAM system based on neural implicit representation, which significantly aids robots in better understanding the scene. FI-SLAM employs a coordinate and plane joint encoding method, which reduces the difficulty of feature storage by flattening the feature space. Furthermore, to improve representation efficiency, we use the method of adjacent feature level linear interpolation to describe features. We propose a feature fusion (FF) method to merge the object features with the scene features. The fused feature vector enhances the reconstruction accuracy of the local scene while ensuring the global reconstruction effect. It has improved the global reconstruction effect of the scene and the accuracy of camera tracking. Numerous experiments on synthetic and real-world datasets demonstrate that our method can assure accurate tracking precision, high-fidelity reconstruction results, and complete semantic instance maps. In summary, the algorithm we proposed heavily augments the scene perception capabilities of robot.**

I. Introduction

Dense semantic simultaneous localization and mapping is an essential challenge in the application of robots [8], [13]. Integrating semantic information into SLAM work can improve the robot's ability to understand the scene, optimize the efficiency of interaction and collaboration, and enhance the accuracy of tracking and mapping.

Traditional visual SLAM algorithms can estimate camera pose with high accuracy and robustness in various scenes [10], [12]. However, they cannot predict unobserved viewpoints, and the storage cost of reconstruction results is high. Recently, Neural Radiance Fields provide a novel solution to deal with these limitations by enabling continuous and dense representation of scenes using limited memory. [8], [16], [18], [21]. In particular, coordinate-based neural networks have received widespread attention for their ability to predict the geometry, shape, and appearance characteristics of any point. Unlike most existing NeRF-SLAM systems, we improve the perceptual ability of the robot by utilizing a joint encoding technique base on coordinates and planes to reconstruction at a semantic instance level. Although some current works have combined semantic and NeRF [19], [20], these tasks require several hours of offline training to obtain a semantic scene, which cannot satisfy the real-time requirements of semantic SLAM.

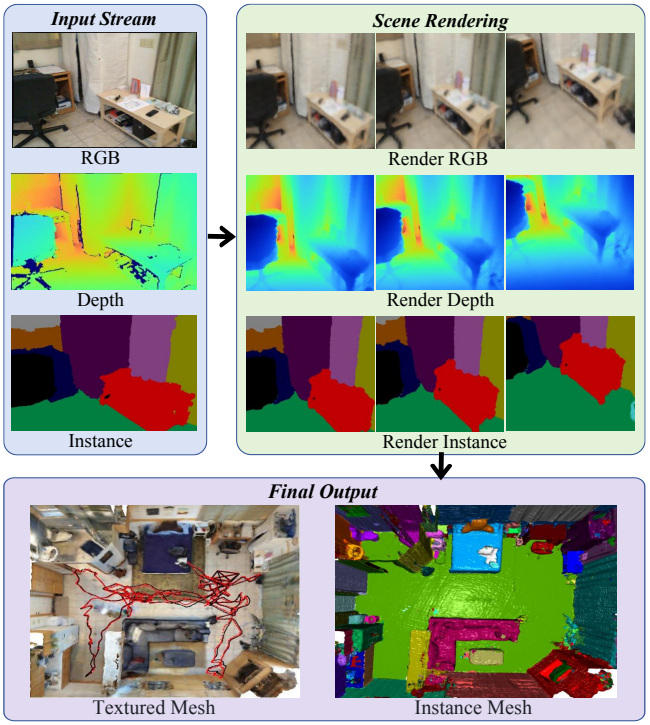

Fig. 1: **The demonstration of FI-SLAM.** We propose FI-SLAM, a dense semantic instance SLAM system based on neural implicit representation, which can provide real-time tracking and mapping.

Semantic-NeRF [19] adds a semantic rendering network after volumetric rendering to reconstruct semantic maps. However, its offline operation makes it impractical for real-time application. NIDS-SLAM [5] accomplishes the real-time construction of semantic maps. However, which relies on the pose provided by ORB-SLAM3 [1] and cannot track independently. Therefore, we propose a feature fusion (FF) method that uses semantic instance and depth projection to obtain object point clouds. Our method can continuously update the point cloud and object bounding box as the camera pose changes. By merging object features and background features, more local features effectively improve the expression of geometry and texture. Meanwhile, we adopted a joint encoding method based on coordinates and planes(CPE) to transform 3D spatial features into three 2D feature planes. Linear interpolation is performed on multiple feature planes to obtain plane features, which are spliced to obtain feature

*The corresponding author of this paper
[1]Xingshuo Wang, Yunzhou Zhang, Zhiyao Zhang, Mengting Wang, Zhiteng Li, Xuanhua Chen are with College of Information Science and Engineering, Northeastern University, Shenyang 110819, China. zhangyunzhou@mail.neu.edu.cn
This work was supported by National Natural Science Foundation of China (No. 61973066) and Major Science and Technology Projects of Liaoning Province(No. 2021JH1/10400049).

vectors. This feature expression method effectively enhances scene reconstruction accuracy and completes defect completion. Our method allows us to get a semantic instance map while improving the accuracy of tracking and reconstruction. Overall, we provide the following contributions:

- We propose FI-SLAM, a dense semantic instance SLAM system based on NeRF. We present a feature fusion (FF) method to describe the local scenes better to obtain scene optimization results.
- We propose a feature fusion (FF) method that uses semantic instance to objectify scene objects and extracts object features and scene features through different feature networks to fusion, effectively improving the expressive ability of geometry and texture in the scene.
- We adopt a joint encoding (CPE) method based on coordinates and planes to transform spatial features into planar features. We obtain feature vectors by performing octilinear interpolation on adjacent level feature planes, for better reconstruction of unobserved viewpoints.
- We evaluate our method with various parameters on two challenging datasets, Replica [15] and ScanNet [3]. Additionally, through ablation experiments, we prove the effectiveness of our algorithm.

II. Related Works

A. Semantic SLAM

Applying instance semantics in the SLAM framework offers numerous advantages. It allows for a more straightforward collection of object-level features, enhances storage efficiency, and provides stability in large-scale changes [4], [14]. All these improvements assist robots in achieving superior self-localization. Meanwhile, it provides a higher level of environmental perception, allowing the robot to perform semantically meaningful tasks more efficiently [9]. SLAM++ [14] uses RGB-D information to represent object scenes via a joint pose graph semantically. Kimera [13] relies on stereo sensors to generate dense semantic meshes and uses visual odometry to estimate poses. Although these methods can perform semantic 3D display reconstruction using depth information, the dense reconstruction may lead to excessive storage space, and low resolution cannot guarantee high fidelity. In this article, we use the small storage costs of neural implicit representations to design a high-precision semantic instance SLAM system, effectively enhancing the robot's environmental perception capabilities.

B. Neural Implicit SLAM

Recently, neural implicit representation has been widely used as an efficient and accurate representation of the entire scene due to its expressiveness and compactness in the reconstruction process. To apply neural implicit representations to the SLAM system, iMAP [17] uses MLP to perform real-time tracking and mapping, introducing key frames strategy to accelerate the operation. NICE-SLAM [21] uses a hierarchical feature network for scene representation, achieving more accurate mapping. Co-SLAM [18] adopts multi-resolution hash encoding, which accelerates the speed of reconstruction and improves the accuracy. All these efforts have proven the feasibility of neural implicit in reconstructing the geometry and color information of the environment. More than this, it can be used to encode semantic information [19]. VMAP [7] uses semantic segmentation results to instantiate different objects. Compared to other methods, our method improves the efficiency and accuracy of tracking and mapping by objectifying the objects in the scene through segmentation results to acquire an additional feature network.

III. Methodology

A. Feature Fusion

Low-rank tensors [2] and hash tables [11] converge quickly. However, they ignore some areas and design pixels as single points and samples, causing excessive blurring in close-ups and sawtooth effects in long shots. Therefore, we propose a feature fusion (FF) reconstruction method. Specifically, we fuse scene and object features during reconstruction to improve the tracking and reconstruction accuracy. Inspired by [6], we adopt a coordinate and plane joint encoding (CPE) method, spheroidizing 3D spatial points and decomposing them into three planes (Fig.3). We get a plane feature vector by performing linear interpolation on two adjacent levels of Mipmap. We concatenate the three plane feature vectors with the coordinate feature vector to get the scene feature vector $\mathbf{f}^s(x)$. To obtain object features, we use instance masks, depth, and predicted camera poses to get object bounding boxes. We transform spatial point coordinates to the object coordinate system to extract object feature vectors $\mathbf{f}_n^o(x)$, where n denotes the instance ID of the object. We fuse the scene and object feature vectors $\mathbf{f}(x) = \mathbf{f}^s(x) + \mathbf{f}_n^o(x)$. The geometric decoder outputs the predicted sdf value and feature vector $\mathbf{h}$:

$$\mathcal{D}_s(\mathbf{f_{sdf}}(x)) \mapsto (sdf, \mathbf{h}) \tag{1}$$

Finally, the predicted color c values and instance values i:

$$\mathcal{D}_c(\mathbf{f_c}(x), \mathbf{h}) \mapsto c \quad \mathcal{D}_i(\mathbf{f_i}(x), \mathbf{h}) \mapsto i \tag{2}$$

B. Object Representation

Bounding Box Acquisition. To extract object features, it is essential to distinguish the representation information of each instance object. We utilize the object mask and depth information to acquire the 3D point cloud of the object. We parameterize the point cloud into bounding box information such as size ε_n, rotation matrix $\mathbf{R}_{ow}^n$, and center μ_n^w through Camera internal parameters and the predicted camera pose. As the number of frames increases, we select keyframes to add to the object, continuously updating the point cloud to optimize the object's bounding box.

Object Coordinate System Conversion. In order to extract object feature $\mathbf{f}_n^o(x)$, the object ID n and coordinates of each sampling point $\mathbf{x}_n^o$ need to be obtained. Therefore, we utilize the bounding box information of the object to calculate them. The sampling points $\mathbf{x}^w$ in Fig.2 and the object center μ_n^w are rotated into the object coordinate system

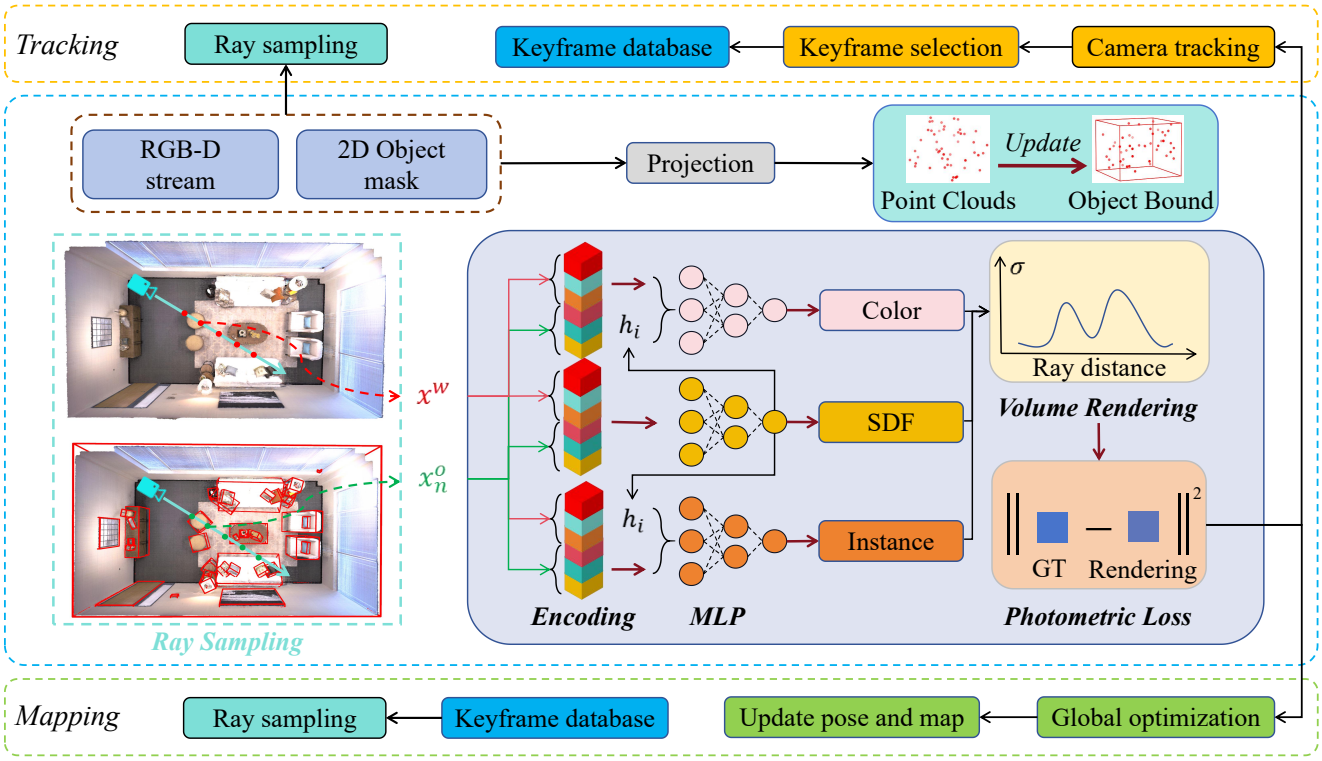

Fig. 2: **An overview of FI-SLAM.** Our method takes RGB-D and semantic instance images as inputs and outputs camera poses and scene representations in the form of jointly optimized feature vectors through a hierarchical feature network. FI-SLAM consists of tracking and mapping, and mapping is performed whenever a keyframe is detected in tracking. During the scene training, FI-SLAM continuously updates the object point cloud through semantic instances and depth information, obtains the bounding boxes of objects, and objectifies space points. Furthermore, during sampling, it normalizes the 3D sampling points in space and inputs them into the object feature network and scene feature network. The output feature vectors are fused to improve scene reconstruction effects and tracking pose accuracy.

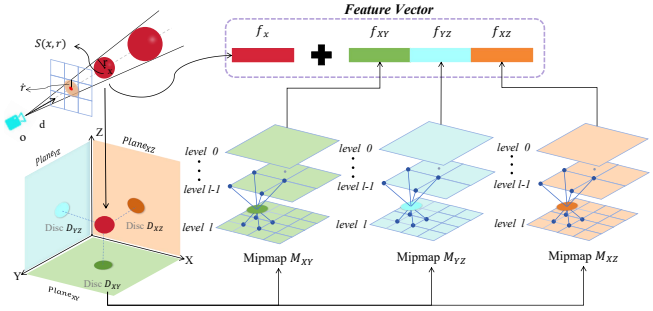

Fig. 3: **Overview of feature encoding.** We project the sampling points in the space into three Discs to flatten the spatial features and reduce the difficulty of feature expression. We perform 8-vertex linear interpolation on two adjacent feature planes in Disc to obtain a planar feature vector. We splice the planar feature vector and the coordinate feature vector to obtain a spatial feature vector.

through the inverse of the rotation matrix $\mathbf{R}_{ow}^{n\ -1}$, which can be written as:

$$\mathbf{x}_n^o = \frac{\mathbf{R}_{ow}^{n\ -1} \cdot (\mathbf{x}^w - \mu_n^w)}{\varepsilon_n} \tag{3}$$

Furthermore, we use the size ε_n to determine the object category of the sampling points and add the object ID to the sampling points, and get the coordinate position of the sampling points in the object coordinate system. To utilize classless sampling points, we set objects such as walls and windows as backgrounds and use the background to constrain these spatial points.

C. Scene Volume Rendering

Volume Rendering. We render depth $\hat{t}$, color $\hat{\mathbf{c}}$ and instance $\hat{\mathbf{i}}$ by integrating predicted values along sampled rays. Specifically, in each frame of the image, we randomly select the same number of background and object pixels using the object mask and choose K sampling points on all the rays $x(t_k) = o + t_k d$ emitted from the center of the projection camera to these pixels, parameterizing the weight of each point on the rays as:

$$w_k = \mathrm{Sigmoid}(\frac{s_k}{tr}) \cdot \mathrm{Sigmoid}(-\frac{s_k}{tr}) \tag{4}$$

where tr is the truncation distance. s_k is the SDF we predict. Each sampled ray has depth values$\{t_1 ... t_K\}$, predicted colors$\{c_1 ... c_K\}$, and predicted instances$\{i_1 ... i_K\}$ and render the color $\hat{\mathbf{c}}$, depth $\hat{t}$ and instance $\hat{\mathbf{i}}$ for each pixel as

$$\hat{\mathbf{c}} = \frac{1}{\sum_{k=1}^{K} w_k} \sum_{k=1}^{K} w_k \mathbf{c_k} \quad \hat{t} = \frac{1}{\sum_{k=1}^{K} w_k} \sum_{k=1}^{K} w_k t_k$$
$$\hat{\mathbf{i}} = \frac{1}{\sum_{k=1}^{K} w_k} \sum_{k=1}^{K} w_k \mathbf{i_k} \tag{5}$$

Objective Functions. We achieve rendering by minimizing the mean squared error between the ground truth and predicted values and updating the camera pose through gradient backpropagation. For effective depth sampling points R, the losses for color and depth are defined as follows:

$$\mathcal{L}_{rgb} = \frac{1}{|R|} \sum_{r \in R} (\hat{\mathbf{c}} - \mathbf{c}_r)^2 \quad \mathcal{L}_t = \frac{1}{|R|} \sum_{r \in R} (\hat{t} - t_r)^2 \tag{6}$$

We use one hot enconding network $\mathbf{H}$ to predict instances. The calculation of the rendering loss for instance reconstruction is as follows:

$$\mathcal{L}_i = \frac{1}{|RN|} \sum_{r \in R} \sum_{n=1}^{N} \mathbf{H}(\mathbf{i}_r^n) log(\mathbf{H}(\hat{\mathbf{i}}^n)) \tag{7}$$

To improve the accuracy of scene geometry reconstruction, we apply truncated signed distance error for sampled depths close to the surface with the truncation region P_r^{tr} where $|t_r - t_p| < tr$. We use the distance between the sample points and the truth value as SDF ground truth for supervision.

$$\mathcal{L}_{sdf} = \frac{1}{|R|} \sum_{r \in R} \frac{1}{|P_r^{tr}|} \sum_{p \in P_r^{tr}} (s_p - (t_r - t_p))^2 \tag{8}$$

For the region P_r^{fs} where is far from the surface with $|t_r - t_p| > tr$, we use a free-space loss to force the SDF to predict the truncation distance tr:

$$\mathcal{L}_{fs} = \frac{1}{|R|} \sum_{r \in R} \frac{1}{|P_r^{fs}|} \sum_{p \in P_r^{fs}} (s_p - tr)^2 \tag{9}$$

The global loss function of our method is defined as:

$$\mathcal{L} = \lambda_{fs}\mathcal{L}_{fs} + \lambda_{sdf}\mathcal{L}_{sdf} + \lambda_t\mathcal{L}_t + \lambda_{rgb}\mathcal{L}_{rgb} + \lambda_i\mathcal{L}_i \tag{10}$$

where $\{\lambda_{fs}, \lambda_{sdf}, \lambda_t, \lambda_{rgb}, \lambda_i\}$ is the weighting factor. It is worth that loss calculations require rays to have actual depth.

D. Tracking and Mapping

Tracking. Our tracking process is a process of minimizing the loss function optimized by the global Adam optimizer. The current estimation of the camera pose is represented by the transformation matrix $\mathbf{T}_{wc} = exp(\xi_t^{\wedge}) \in \mathbb{SE}(3)$. We initialize the pose of the current frame t through a constant speed assumption, which can be designed as:

$$\mathbf{T}_t = \mathbf{T}_{t-1}\mathbf{T}_{t-2}^{-1}\mathbf{T}_{t-1} \tag{11}$$

Mapping. Our mapping is the process of scene representation based on feature fusion (FF) encoder and MLP decoder. We choose K pixels from M frames, which include the current frame, the previous two keyframes, and $M-3$ frames randomly selected from the keyframe list. We jointly optimized the scene features, object features, and MLP decoder of M frames by minimizing the overall loss function. We adopt a method of joint encoding (CPE) based on coordinates and planes to fill in the unobserved viewpoints of scene, improving the completeness of scene.

IV. Experiments

A. Experimental Setup

Datasets and evaluations. We evaluate the performance of FI-SLAM on eight scenes on the simulated dataset Replica [15] and six scenes on the real-world dataset ScanNet [3]. For reconstruction quality indicators, we use Depth L1, Accuracy (Acc.), Completion (Comp.), and Completion ratio (Comp. Ratio) with a threshold of $5cm$. For pose prediction, we use Absolute Trajectory Error (ATE) to evaluate tracking accuracy on Replica [15] and ScanNet [3].

Baseline. Current datasets and algorithms are unable to provide a semantic instance reconstruction map, so we present a qualitative illustration of 2D rendered images of the scene and 2D ground truth images. For tracking accuracy and reconstruction effect, we use iMAP [16], NICE-SLAM [21], and Co-SLAM [18] as our comparative baselines, where iMAP* [16] indicates the experimental results of iMAP [16] by the author of NICE-SLAM [21].

Implementation Details. We run FI-SLAM on a desktop PC with an Intel Core i7-11700 (16 cores @ 2.50GHz), 32GB of RAM, and a single NVIDIA GeForce RTX 3080 GPU. FI-SLAM leverages a 24-channel feature vector to represent semantics, geometry, and texture. During the sampling process, 1024 pixels are selected from the object and the background. The tracking process involves 15 iterations, while the mapping process involves 20 rounds of iterations. Every 5th frame we add a scene keyframe. Within the object feature network, the background resolution is 256^3, and the object resolution is 64^3. In the scene feature network, the scene resolution is set at 512^3. When creating object instance, 15 keyframes are retained for each object.

B. Experimental results

Replica Dataset. To evaluate the reconstruction accuracy in Replica [15], we carry out assessments on the same simulated RGB-D sequence as with Co-SLAM [18]. The Depth L1, Acc, Comp, and Comp. Ratio are used as evaluation indicators. As shown in Tab.I, our method is almost as accurate as Co-SLAM [18], and we have higher completeness.

TABLE I: **Quantitative results for reconstruction in the eight scenes of Replica [15].** We report the Depth L1[cm]↓, Acc.[cm]↓, Comp.[cm]↓, Comp. Ratio[$< 5cm\%$]↑ for reconstruction in the eight scenes of Replica [15]. Additionally, we calculate the average values of the fourth metric.

Method	Depth L1↓	Acc↓	Comp↓	Comp.Ratio↓
iMAP* [16]	4.64	3.62	4.93	80.50
NICE-SLAM [21]	1.90	2.37	2.63	91.13
Co-SLAM [18]	1.51	**2.10**	2.08	93.44
Ours	**1.33**	2.26	**1.72**	**96.31**

NICE-SLAM [21] employs a dense grid representation, preserving more reconstruction details, but artifacts are present in local details. Co-SLAM [18] utilizes a multi-resolution hash, enabling better detail description. Still, its reconstruction precision is low, and the fulfillment effect is unsatisfactory. Our method enhances local appearance fidelity by introducing object representation, while our feature fusion ensures the geometric consistency of the scene. As shown in Fig.4, in local scene reconstruction, such as the completion of the area behind the sofa in room-0 and the reconstruction of the table edge in office-3, our method performs better in reconstruction and completion compared to other algorithms.

Fig.5 exhibits the instance scene that we have constructed. Due to the omission of ground truth for instance scenes in the Replica dataset, we use network-rendered 2D instance images and truth images for qualitative illustration. The tagging mapping methods we employ in 3D instance reconstruction and 2D instance rendering differ. Therefore, the colors vary

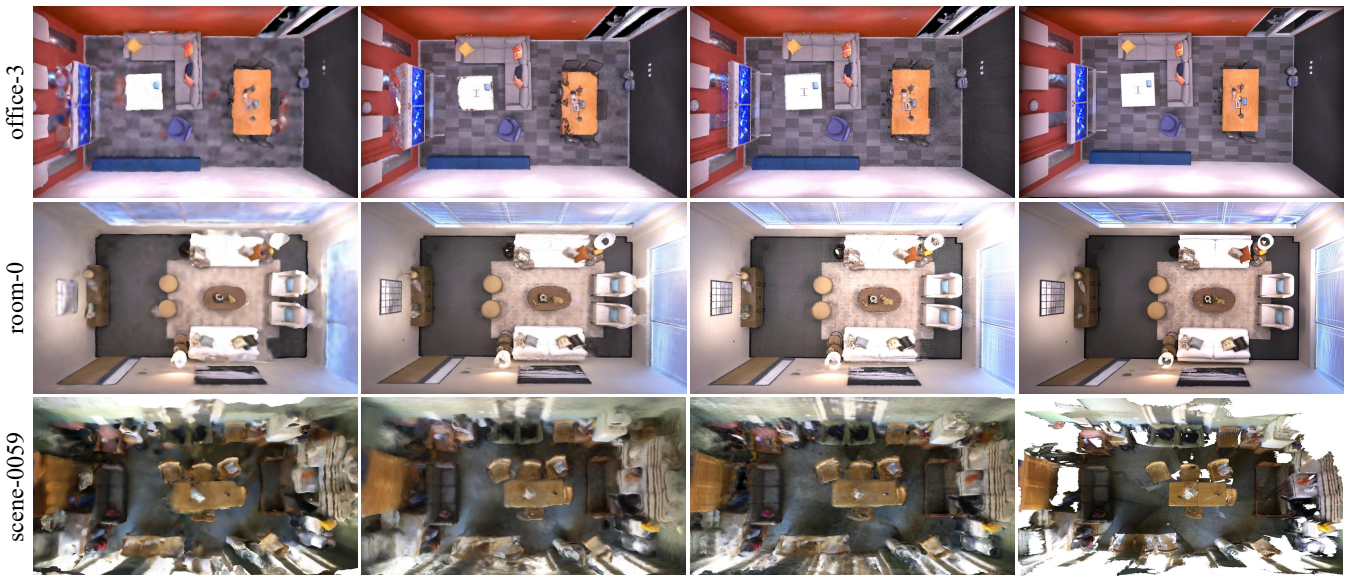

Fig. 4: **Reconstruction results on two datasets.** Compared with existing methods, our method achieves high-fidelity scene reconstruction in various scenarios and more effectively completes the unobserved viewpoints.

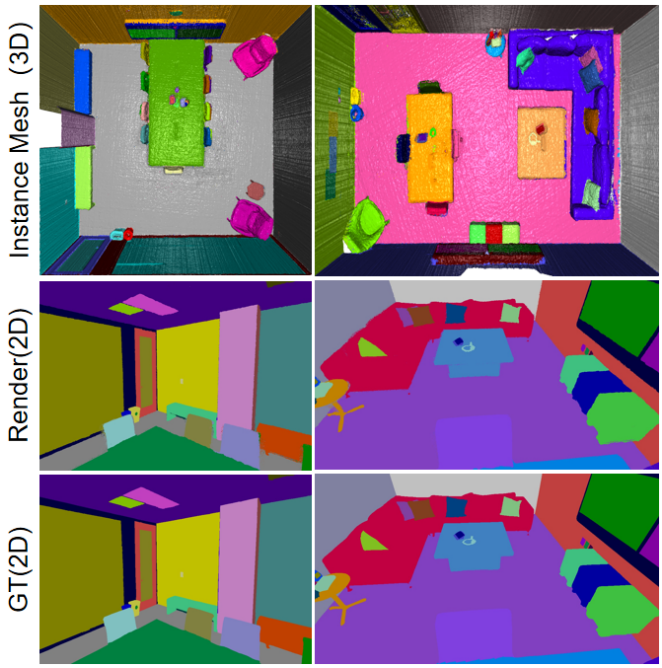

Fig. 5: **Reconstruction results of semantic instances on Replica [15] dataset.** Replica [15] dataset cannot provide ground truth for instance reconstruction, so we qualitatively compare the rendered 2D instance image and the ground truth instance image. The color labels we use while carrying out 3D reconstruction and 2D rendering are different, so the colors in the image differ.

in the images. It is not due to prediction errors that the colors differ.

ScanNet Dataset. To evaluate the tracking accuracy of FI-SLAM in the ScanNet [3] dataset, we compare the ATE RMSE of our estimated camera poses with other methods. As shown in Tab.II, on average, our method demonstrate the best performance in ATE RMSE, improving the tracking accuracy by 2cm compared to existing algorithms. In the scene-0106 sequence, due to the lack of 20 frames of instance segmentation images, 20 frames cannot be trained, leading to poor tracking accuracy. However, our tracking accuracy is higher than existing methods in other sequences.

TABLE II: **The ATE RMSE↓ [cm] result run on ScanNet [3].** We report the ATE RMSE↓ [cm] for camera tracking in the six scenes. FI-SLAM achieves better tracking accuracy than existing algorithms, with an average improvement of above **2cm** over other methods.

Method	0000	0059	0106	0169	0181	0207	Avg.
iMAP* [16]	55.95	32.06	17.50	70.51	32.10	11.91	36.67
NICE-SLAM [21]	8.64	12.25	**8.09**	10.28	12.93	5.59	9.63
Co-SLAM [18]	7.18	12.29	9.57	6.62	13.43	7.13	9.37
Ours	**6.50**	**8.67**	9.98	**4.87**	**8.57**	**5.42**	**7.23**

We provide a qualitative analysis of camera tracking and reconstruction effects in Fig.**??**. The results show that our method is not affected by large offsets and more stable than existing methods. The ground truth scene in the ScanNet [3] dataset is incomplete, so we can only provide a qualitative analysis of the geometric reconstruction of the real-world ScanNet [3] dataset. The qualitative comparison experiment in Fig.4 verifies that our algorithm has better reconstruction effects compared to existing methods.

C. Ablation Studies

Effectiveness of FF. As shown in Fig.6, we have used different feature networks for training. During the reconstruction of the object part, the Object Features (OF) will result in jagged edges and cannot perform texture constraints. At the same level of the network, Scene Features (SF) will result in excessive smoothing when reconstructing as a whole. In contrast, our Feature Fusion (FF) algorithm effectively alleviates the above two situations and improves the overall reconstruction accuracy of the scene. Tab.III shows the ATE RMSE in the six scenes of the Replica [15] dataset

under three different feature algorithms. Feature fusion(FF) algorithm provides more accurate tracking accuracy.

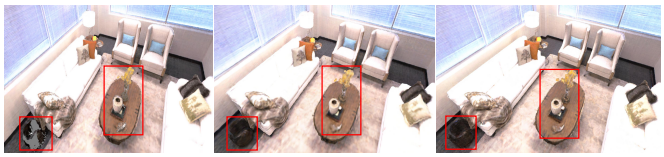

Fig. 6: **Effectiveness of FF.** We demonstrated the reconstruction in room-0 using OF, SF, and FF as scene representations. OF would exhibit texture distortion and geometric mutations during the reconstruction, while SF would produce an excessively smoothed scene. FF has better geometry and texture compared to OF and SF.

TABLE III: **Quantitative tracking results for FF ablation.** We present the tracking ATE RMSE↓ $[cm]$ of OF, SF, and FF in six scenarios on the Replica [15] dataset.

Method	room0	room1	room2	office2	office3	office4	Avg.
OF	0.87	1.16	1.06	1.71	2.20	0.89	1.32
SF	0.70	0.97	1.03	1.37	1.03	1.05	1.03
FF	**0.62**	**0.68**	**0.97**	**0.75**	**0.88**	**0.88**	**0.80**

Effectiveness of CPE. Fig.7 shows the results we achieved in the room-1 of the Replica [15] dataset by adopting Plane Encoding (PE) and Coordinate and Plane Joint Encoding (CPE). Intuitively, it can be seen that the plane encoding still has gaps in the completion of unobserved viewpoints. Incorporating coordinate coding for joint encoding has enhanced the ability to fill in the blanks in unobserved viewpoints.

Fig. 7: **Effectiveness of CPE.** We demonstrated the reconstruction in room-1 using PE and CPE as scene encoding. CPE achieves more efficient completion effects than PE for unobserved viewpoints.

V. Conclusion

FI-SLAM is a dense semantic SLAM system based on neural implicit representation. We adopt a coordinate and plane joint encoding method to represent the scene and use feature fusion for training. A large number of experiments show that our method achieves high-fidelity mapping and accurate tracking and also completes the scene from unobserved viewpoints to a large extent. In semantic instance SLAM, we provide high-fidelity instance maps, effectively enhancing the robot's perception of the environment.

References

[1] Carlos Campos, Richard Elvira, Juan J. Gomez Rodriguez, Jose M. M. Montiel, and Juan D. Tardos. Orb-slam3: An accurate open-source library for visual, visual-inertial and multi-map slam. *IEEE Transactions on Robotics*, page 18741890, Dec 2021.

[2] Anpei Chen, Zexiang Xu, Andreas Geiger, Jingyi Yu, and Hao Su. Tensorf: Tensorial radiance fields. In *European Conference on Computer Vision*, pages 333–350. Springer, 2022.

[3] Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5828–5839, 2017.

[4] Michael Elad, TylerH. Summers, Tony Wood, Chris Manzie, and Iman Shames. Probabilistic data association for semantic slam at scale. *Cornell University - arXiv,Cornell University - arXiv*, Feb 2022.

[5] Yasaman Haghighi, Suryansh Kumar, JeanPhilippe Thiran, and Luc-Van Gool. Neural implicit dense semantic slam. Apr 2023.

[6] Wenbo Hu, Yuling Wang, Lin Ma, Bangbang Yang, Lin Gao, Xiao Liu, and Yuewen Ma. Tri-miprf: Tri-mip representation for efficient anti-aliasing neural radiance fields. Jul 2023.

[7] Xin Kong, Shikun Liu, Marwan Taher, and AndrewJ. Davison. vmap: Vectorised object mapping for neural field slam. Feb 2023.

[8] John McCormac, Ankur Handa, Andrew Davison, and Stefan Leutenegger. Semanticfusion: Dense 3d semantic mapping with convolutional neural networks. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, May 2017.

[9] IanD. Miller, Fernando Cladera, Trey Smith, CamilloJose Taylor, and Vijay Kumar. Stronger together: Air-ground robotic collaboration using semantics. Jun 2022.

[10] Raul Mur-Artal and Juan D. Tardos. Orb-slam2: an open-source slam system for monocular, stereo and rgb-d cameras. *IEEE Transactions on Robotics*, page 12551262, Oct 2017.

[11] Thomas Mller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM Transactions on Graphics*, page 115, Jul 2022.

[12] Richard A. Newcombe, Steven J. Lovegrove, and Andrew J. Davison. Dtam: Dense tracking and mapping in real-time. In *2011 International Conference on Computer Vision*, Nov 2011.

[13] Antoni Rosinol, Marcus Abate, Yun Chang, and Luca Carlone. Kimera: an open-source library for real-time metric-semantic localization and mapping. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, May 2020.

[14] Renato F. Salas-Moreno, Richard A. Newcombe, Hauke Strasdat, Paul H.J. Kelly, and Andrew J. Davison. Slam++: Simultaneous localisation and mapping at the level of objects. In *2013 IEEE Conference on Computer Vision and Pattern Recognition*, Jun 2013.

[15] Julian Straub, Thomas Whelan, Lingni Ma, Yufan Chen, Erik Wijmans, Simon Green, Jakob J. Engel, Raul Mur-Artal, Carl Ren, Shobhit Verma, Anton Clarkson, Mingfei Yan, Brian Budge, Yajie Yan, Xiaqing Pan, June Yon, Yuyang Zou, Kimberly Leon, Nigel Carter, Jesus Briales, Tyler Gillingham, Elias Mueggler, Luis Pesqueira, Manolis Savva, Dhruv Batra, Hauke M. Strasdat, Renzo De Nardi, Michael Goesele, Steven Lovegrove, and Richard Newcombe. The Replica dataset: A digital replica of indoor spaces. *arXiv preprint arXiv:1906.05797*, 2019.

[16] Edgar Sucar, Shikun Liu, Joseph Ortiz, and Andrew J Davison. imap: Implicit mapping and positioning in real-time. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6229–6238, 2021.

[17] Edgar Sucar, Shikun Liu, J.V. Ortiz, and AndrewJ. Davison. imap: Implicit mapping and positioning in real-time. *International Conference on Computer Vision,International Conference on Computer Vision*, Jan 2021.

[18] Hengyi Wang, Jingwen Wang, and Lourdes Agapito. Co-slam: Joint coordinate and sparse parametric encodings for neural real-time slam. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13293–13302, 2023.

[19] Shuaifeng Zhi, Tristan Laidlow, Stefan Leutenegger, and Andrew J. Davison. In-place scene labelling and understanding with implicit scene representation. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, Oct 2021.

[20] Shuaifeng Zhi, Edgar Sucar, Andre Mouton, Iain Haughton, Tristan Laidlow, and AndrewJ Davison. ilabel: Revealing objects in neural fields.

[21] Zihan Zhu, Songyou Peng, Viktor Larsson, Weiwei Xu, Hujun Bao, Zhaopeng Cui, Martin R. Oswald, and Marc Pollefeys. Nice-slam: Neural implicit scalable encoding for slam. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12776–12786, 2022.